A Multi-Perspective View on Model-Based Exceptional Link Analysis on Complex Interaction Networks

Atzmueller, Martin; Güven, Çiçek

Publication date:
2019

Citation for published version (APA):
Atzmueller, M., & Güven, Ç. (2019). *A Multi-Perspective View on Model-Based Exceptional Link Analysis on Complex Interaction Networks*. Poster session presented at Annual Conference of The Netherlands Platform of Complex Systems, Utrecht, Netherlands.

A Multi-Perspective View on Model-Based Exceptional Link Analysis on Complex Interaction Networks

Martin Atzmueller, Cicek Güven

Tilburg University (CSAI)

http://www.vikamine.org

Abstract

The detection of anomalies and exceptional patterns in the context of link analysis on complex interaction networks is a prominent research direction in complexity and network science. Applications include, e.g., fraud detection in online social networks, discovering events or unusual topics in heterogeneous network data, or identifying specific interesting or outstanding behavior, for example, considering influential or "central" actors. Taking an abstract view, an anomaly can be considered as a pattern that does not conform to some notion of the expected, normal behavior. However, there is usually no clear formalization of the "normal behavior". In addition, the notion of an anomaly includes other factors compared to a mere outlier which is typically defined by statistical criteria. The concept of an anomaly typically captures more complex criteria, including semantics, (user) expectations and complex data-driven structures. Thus, it is difficult to formalize anomalies with a complex structure, e.g., relating to a group structure instead of considering isolated points. Therefore, such complex (collective) anomaly patterns are often not detected if the individual contained points seem normal and only their interaction causes an anomaly. In addition, the complexity of anomaly detection is further enhanced by multi-relational and multi-dimensional data.

Example Application: Openbeacon wearable sensor for *smart social sensing*, i.e., for the detection of face-to-face contacts (http://www.sociopatterns.org).
Constructing offline social networks based on human face-to-face proximity.

Model-Based Options

Explicit Formalization

Compilation

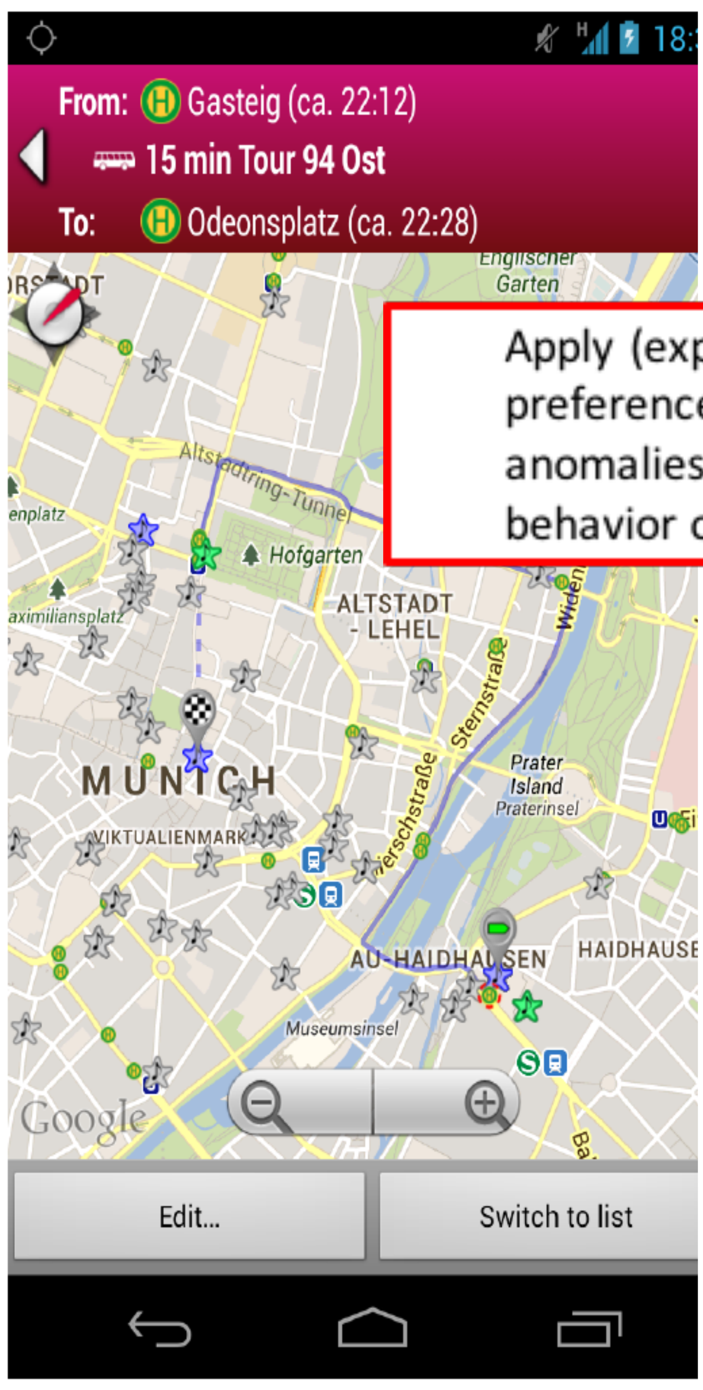
Data-/Structure-driven

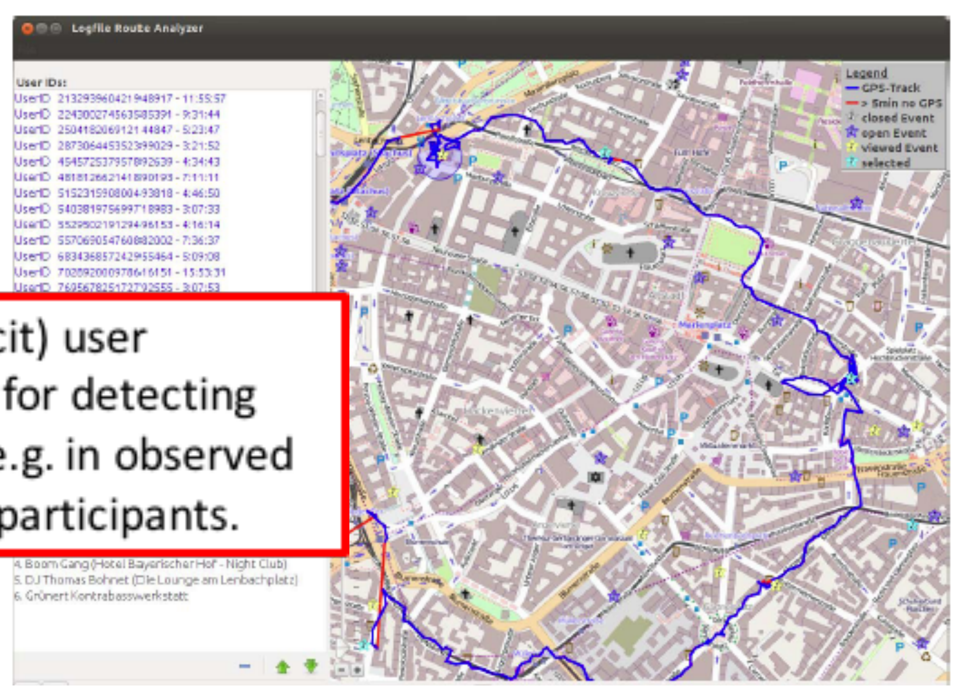
Theory-Based Modeling
Knowledge-Based Modeling
Preference-Based Modeling

Behavior-Based Modeling
Structure-Based Modeling

Preference-Based Modeling

- Social Event Network Analysis
- Context: Music Event
  - Preference(planned performances)
- Attendance (GPS tracks)
- Analysis of structure/behavior
  - On Graphs & Projections
  - Mining exceptional patterns (i.e. deviations/anomalies)





Apply (explicit) user preferences for detecting anomalies, e.g. in observed behavior of participants.

Behavior-/Structure-Based Modeling Examples:

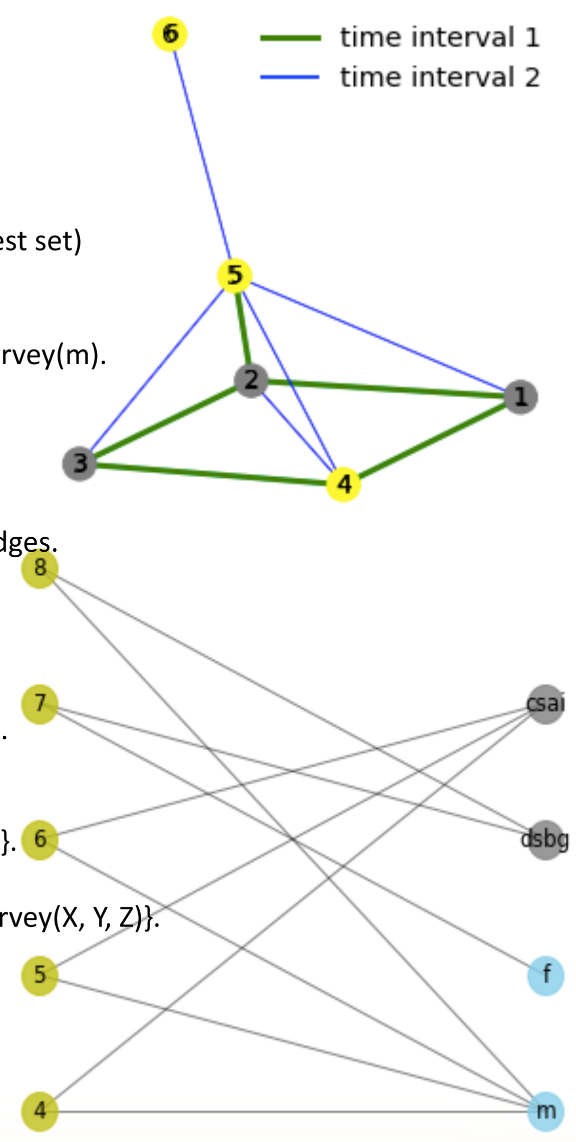
- Interestingness measure: large patterns with large deviation, e.g. comparison to null model
- Specific graph structures, e.g. cliques, k-cores, ...
- Also: Combination of the above, e.g. for mining local exceptional patterns.

Knowledge-Based Modeling

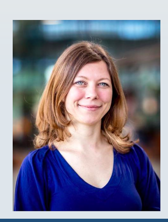
Answer Set Programming

```
#const n=2.
#const n_survey=2.
% ASP facts - defining the networks/graphs
node(1..6). % Nodes of the interaction graph
edge(1, 2).  edge(1, 4). edge(2, 5). edge(2, 3). edge(3, 4). % Edges, first time interval
test(2, 4). test(1, 5). test(3, 5). test(4, 5). test(5, 6).  % Edges, second time interval (test set)

% Nodes and edges of the survey graph:
node_survey(4..8). node_survey(csai). node_survey(dsbg). node_survey(f). node_survey(m).
edge_survey(5, csai). edge_survey(8, dsbg). edge_survey(7, dsbg).
edge_survey(4, csai). edge_survey(6, csai). edge_survey(5, m). edge_survey(4, m).
edge_survey(8, m). edge_survey(7, f). edge_survey(6, m).
% ASP rules
edge(Y, X) :- edge(X, Y). % This is an undirected graph,  hence there is symmetry in edges.
edge_survey(Y, X) :- edge_survey(X, Y).

% X is a common neighbor of Y and Z where they are not connected.
c(X, Y, Z) :- edge(X, Y), edge(X, Z),  not edge(Y, Z), Y!=Z.
c_survey(X, Y, Z) :- edge_survey(X, Y), edge_survey(X, Z),  not edge_survey(Y, Z), Y!=Z.

% a link is predicted when there are 2 common neighbors in the interaction graph
cn_lp_interaction(Y, Z) :- node(Y), node(Z), not edge(Y, Z), Y!=Z,  n=#count{X:c(X, Y, Z)}.
% a link is predicted  when there are 2 common neighbors in the survey graph
cn_lp_survey(Y, Z) :- node(Y), node(Z), not edge(Y, Z), Y!=Z,  n_survey=#count{X:c_survey(X, Y, Z)}.

test(Y, X) :- test(X, Y).
% The match rule compares the predicted set of links with the test set
match(X, Y) :- test(X, Y),  cn_lp_interaction(X, Y).
match(X, Y) :- test(X, Y),  cn_lp_survey(X, Y).

% interactions predicted by interaction network, predicted by  survey, but not in the test set
anomaly1(X,Y):-cn_lp_interaction(X, Y),cn_lp_survey(X, Y), not test(X, Y).
% interactions NOT predicted by interaction network, predicted by  survey network, but not in the test set
anomaly2(X,Y):- not cn_l
% interactions NOT predi
anomaly3(X,Y):- not cn_l

#show anomaly1/2.
#show anomaly2/2.
#show anomaly3/2.
```

```
clingo version 5.3.0
Reading from stdin
Solving...
Answer: 1
anomaly2(4,6) anomaly2(6,4) anomaly3(5,6) anomaly3(5,4) anomaly3(6,5) anomaly3(4,5)
SATISFIABLE
```



Martin Atzmueller
Tilburg University
m.atzmuller@uvt.nl



Cicek Güven
Tilburg University
c.guven@uvt.nl